INSTITUTE FOR DEFENSE ANALYSES

DATAWorks 2023 – Introduction to Design of Experiments in R: Generating and Evaluating Designs with skpr

Tyler T. Morgan-Wall, Project Leader

April 2023

IDA Document NS D-33397

Log: H 2023-000047

INSTITUTE FOR DEFENSE ANALYSES
730 East Glebe Road
Alexandria, Virginia 22305

IDA

The Institute for Defense Analyses is a nonprofit corporation that operates three Federally Funded Research and Development Centers. Its mission is to answer the most challenging U.S. security and science policy questions with objective analysis, leveraging extraordinary scientific, technical, and analytic expertise.

About This Publication
This work was conducted by the Institute for Defense Analyses (IDA) under contract HQ0034-19-D-0001, Task C9117, “Design of Experiments in R: Tutorial". The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

For more information:
Dr. Tyler T. Morgan-Wall, Project Leader
tmorganw@ida.org • (703) 845-2095

Dr. V. Bram Lillard, Director, Operational Evaluation Division
vlillard@ida.org • (703) 845-2230

Rigorous Analysis | Trusted Expertise | Service to the Nation

INSTITUTE FOR DEFENSE ANALYSES

IDA Document NS D-33397

DATAWorks 2023 – Introduction to Design of Experiments in R: Generating and Evaluating Designs with skpr

Tyler T. Morgan-Wall, Project Leader

Executive Summary

This workshop instructs attendees on how to run an end-to-end optimal Design of Experiments (DOE) workflow in R using the open-source skpr package. This workshop is split into three sections: basic R usage, optimal design generation, and design evaluation.

The workshop first provides basic instructions on how to use R, covering how to install packages, create different data structures, use loops, and ensure reproducibility by setting a random seed. This short introduction to R gives an attendee the minimum set of skills they will need to use skpr in their DOE workflow.

The workshop then explains how to use skpr to generate experimental designs, providing information on the various types of designs that can be generated using skpr (e.g., D-optimal, I-optimal, and Alias-optimal designs) as well as the experimental goals that each type of design is intended to best address (e.g., characterization, prediction, or screening).

The final segment of the workshop covers design evaluation with skpr: how to determine whether an experimental design is adequate for the test at hand. One of the primary ways to assess design quality is statistical power, which is a measure of whether a design can detect an effect if one truly exists. The workshop provides information on how to perform statistical power calculations and how to use these calculations to efficiently allocate test resources. However, all power calculations are valid only if the tester uses specific analytical methods when analyzing their data after the test has been executed. Thus, attendees also learn how their power analysis informs their actual final data analysis.

Contents

Design of Experiments in R with the “skpr” Package

Tyler Morgan-Wall

April 26, 2023

Institute for Defense Analyses
730 East Glebe Road ● Alexandria, Virginia 22305-3086

What you'll learn from this talk

- The basics of using R (installing software/writing scripts/reading and writing data) and working in a reproducible workflow.
- How to generate optimal designs entirely in code using **skpr**.
- How to use **skpr**'s design evaluation interface to easily calculate power for linear models as well as non-linear models (e.g., logistic regression).
- How your power analysis informs your actual data analysis.
- How to generate power versus sample size curves in only a few lines of code.

Outline

1. Introduction to Design of Experiments/R/skpr
2. Optimal Design Generation
3. Evaluating Statistical Power
4. skprGUI

Outline

1. Introduction to Design of Experiments/R/skpr
2. Optimal Design Generation
3. Evaluating Statistical Power
4. skprGUI

Why use R?

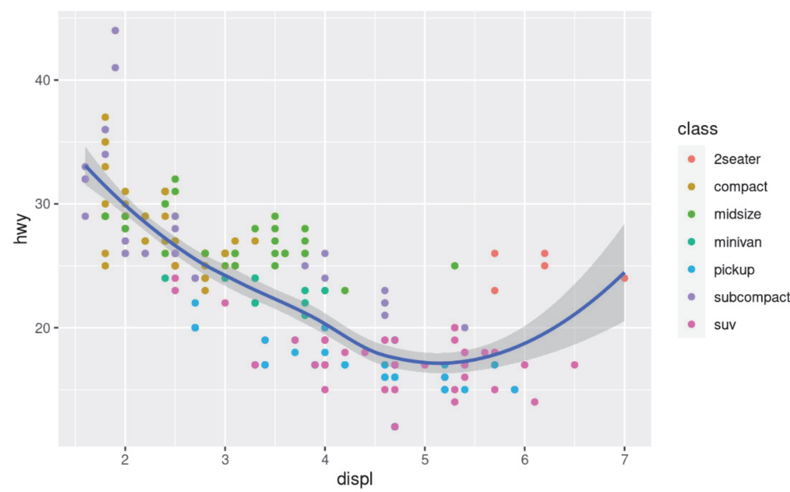
Why use DOE?

Why use optimal DOE?

Why use skpr for optimal DOE?

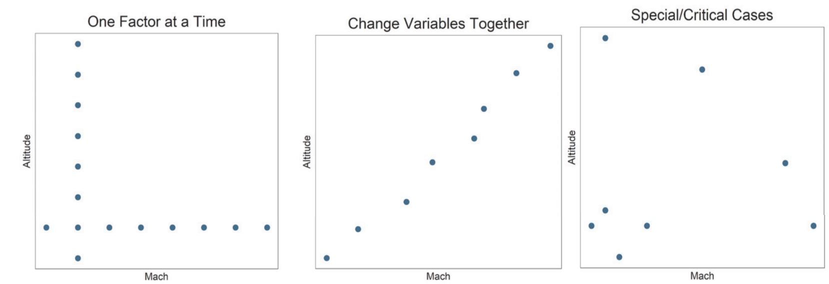
Intro: Why use R?

- Purpose-built for statistics: R is a programming language that was designed to manipulate and analyze data.
- Reproducibility: Scripting your analysis can reduce human error and allow you to revisit and re-use old analyses for new projects.
- Unparalleled statistical capabilities: R has a massive repository of mature, well-documented software libraries.
- Open source: It’s free and cross-platform (Windows/Linux/macOS).

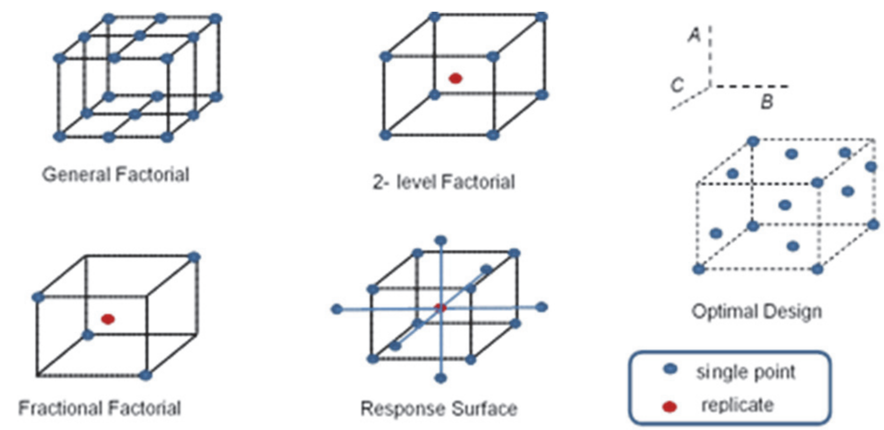


Intro: Why use Design of Experiments?

- Design of Experiments (DOE) is a branch of applied statistics that deals with planning, conducting, analyzing, and interpreting controlled tests.
- DOE enables you to design tests that investigate multiple factors in one experiment (i.e., avoiding inefficient “one factor at a time” testing).
- DOE provides the tester with an analytical framework to determine whether a test is good enough for their purpose (e.g., model selection, estimating effect sizes, optimizing the value of a response).
- A test strategy that employs DOE will provide the most power allocation of test resources for a given number of events.



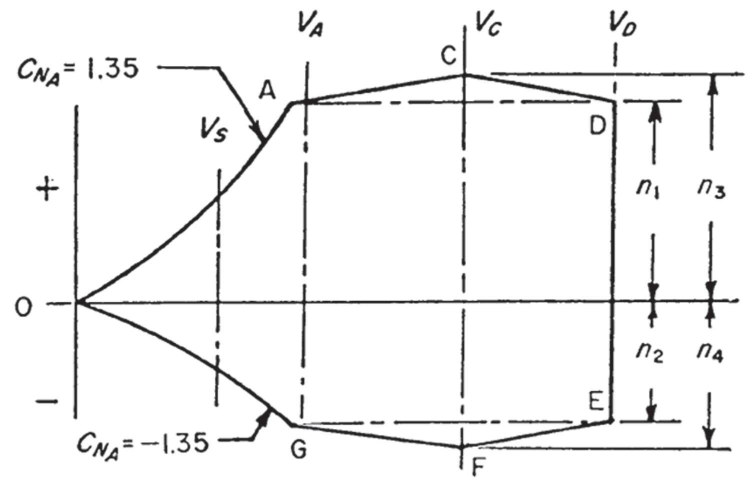
Ad-Hoc Testing
vs
Design of Experiments



Intro: Why use Optimal Design of Experiments?

- Traditional DOE designs are ideal for test programs that have flexibility on amount of test resources required and little to no test execution restrictions.
- But many real-world tests include non-ideal constraints.
  - o Example: Limited resources
  - o Example: Disallowed combinations
- Optimal DOE allows you algorithmically generate a design that is customized for your specific testing needs.

FIGURE A-4—FLIGHT ENVELOPE



1. Conditions "C" or "F" need only be investigated when $n_3 \frac{W}{S}$ or $n_4 \frac{W}{S}$ is greater than $n_1 \frac{W}{S}$ $W|_{40}$, respectively.

2. Condition "G" need not be investigated when the supplementary condition specified in § 23.869 is investigated.

What is an optimal design?

- Classical DOE: A subset of optimal designs that can be found using analytic methods
- Ideal for experiments with no resource/factor constraints: many desirable properties
- Examples: Full Factorial, Fractional Factorial, Central Composite
- *Downside: Limited flexibility*

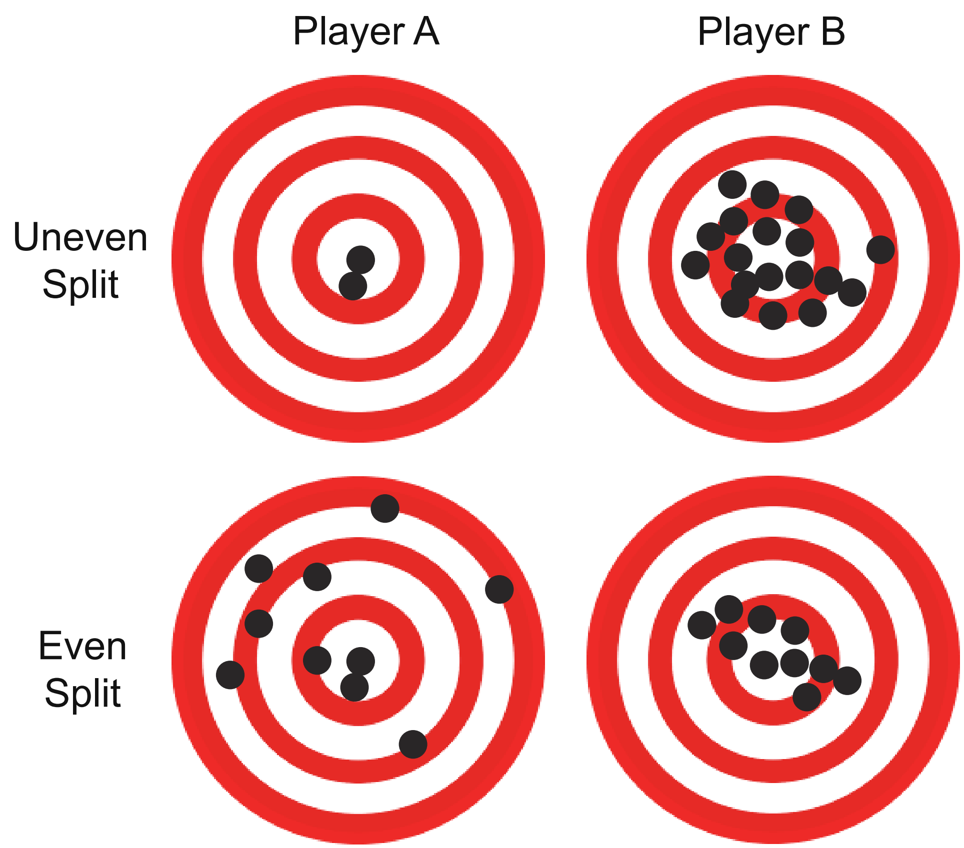
Classical DOE

Optimal DOE

- Optimal DOE: Designs that optimize a particular desirable statistical criterion
- Versatile and efficient: Adapts to real-world test conditions to find “best” design given constraints
- Examples: D-optimal, I-optimal, Alias-Optimal designs
- *Downside: Must be found algorithmically*

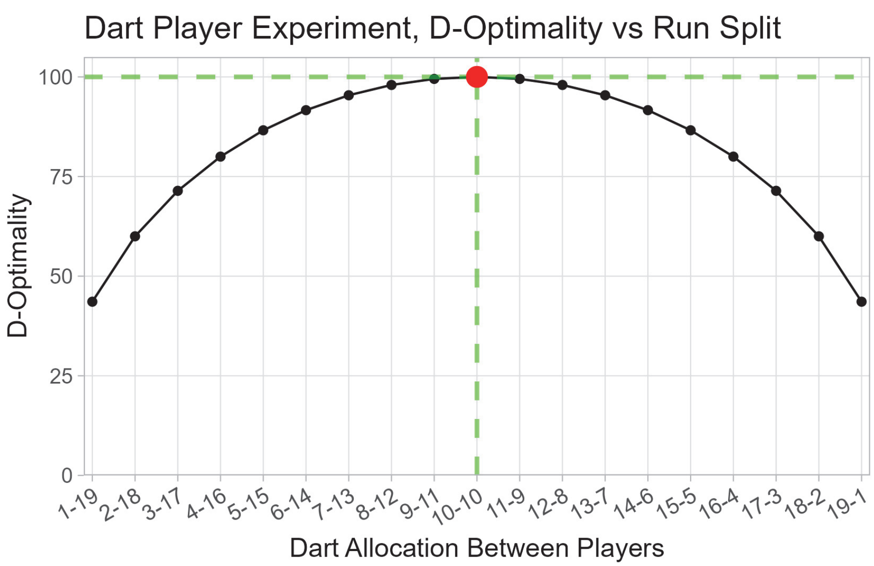
Example: Comparing the performance of two dart players

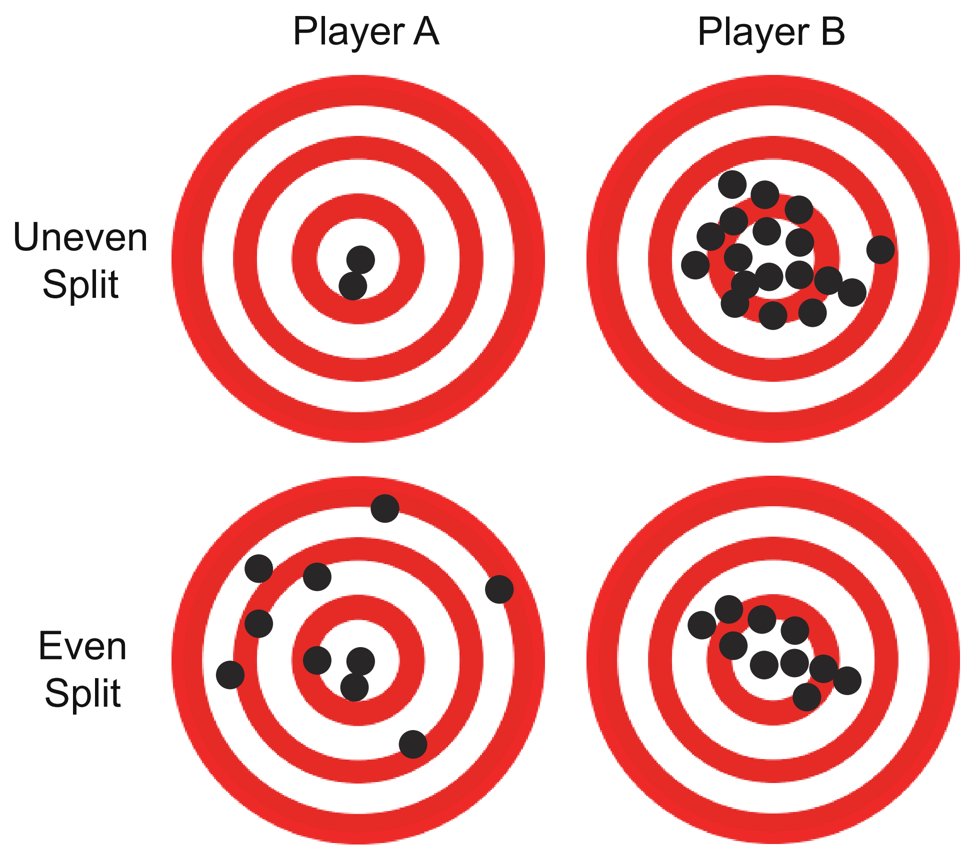
We have 20 darts to distribute: what's the best way to allocate them between the two players to estimate who plays better?



Example: Comparing the performance of two dart players

We have 20 darts to distribute: what's the best way to allocate them between the two players to estimate who plays better?



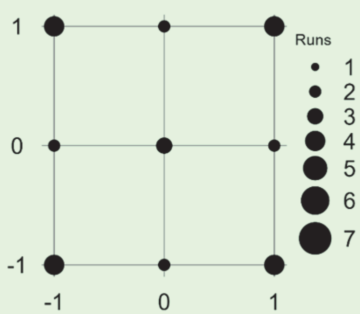


Relevant Optimality Criteria Overview

D-Optimal

- D-optimal designs minimize the variance of the parameter estimates
- Most useful for characterizing performance
- Generally, provides the “best” power across all parameter terms
- Tends to place most points on the edges of the design space

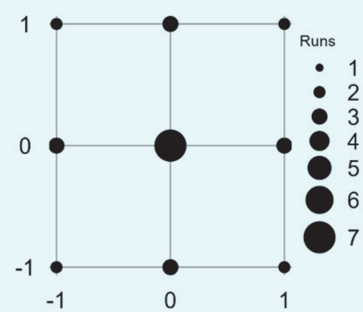
Example:
2 factor model with quadratic terms and all interactions



I-Optimal

- I-optimal designs minimize the average prediction variance
- Most useful when you want good predictions across your design space, particularly if you want to model curvature
- Tends to allocate more points on the interior

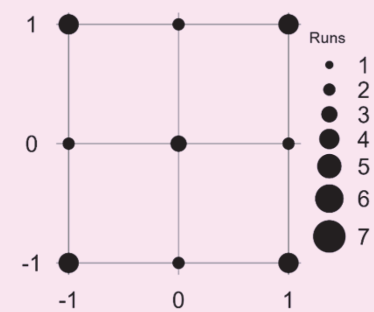
Only difference here in inputs is design is I-optimal—resulting in more points allocated in the interior.



Alias-Optimal

- Alias-optimal designs minimize correlation between main effect and interaction terms
- Most useful when performing screening experiments
- Usually less power to estimate effect sizes, but can reduce overall resource requirements with sequential testing

Note: Here, the Alias-optimal design is the same as D-optimal design!



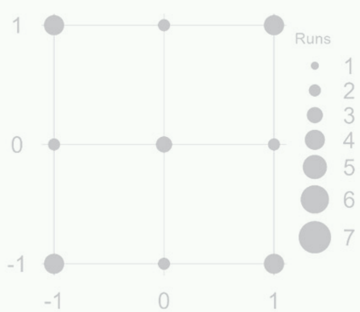
Design Search Computational Difficulty

D-Optimal

- D-optimal designs minimize the variance of the parameter estimates
- Most useful for characterizing performance
- Generally, pr... the “bes... all para...
- Tends ... points on the ...ges of the design space

Low

Example:
2 factor model with quadratic terms and all interactions



I-Optimal

- I-optimal designs minimize the average prediction variance
- Most useful when you want good predictions across your design s... parti...
- ...ore ... the interior

Medium

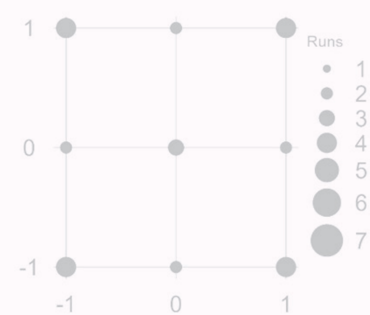
Only difference here in inputs is design is I-optimal—resulting in more points allocated in the interior.

Alias-Optimal

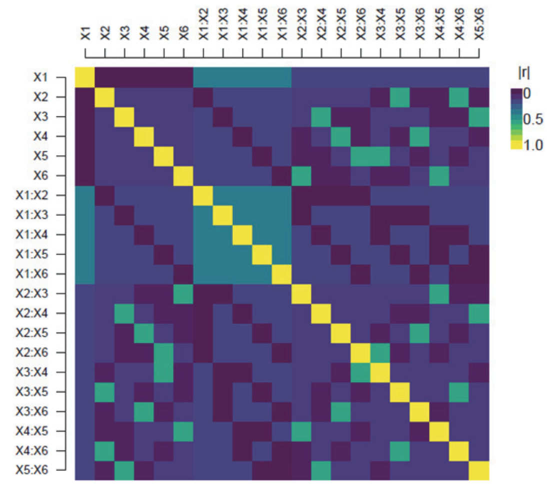
- Alias-optimal designs minimize correlation between main effect and interaction terms
- Most useful when performing screening exp...s
- Usu... estim...ut can reduc...ce require...nts with sequential testing

High

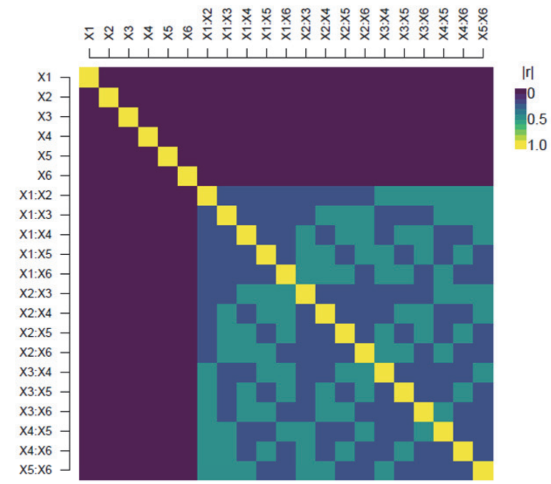
Note: Here, the Alias-optimal design is the same as D-optimal design!



Example: Alias vs D-optimal



(a) Correlation matrix for a D-optimal design.



(b) Correlation matrix for an Alias-optimal design.

Parameter	Power
(Intercept)	0.969
X1	0.969
X2	0.969
X3	0.969
X4	0.969
X5	0.969
X6	0.969

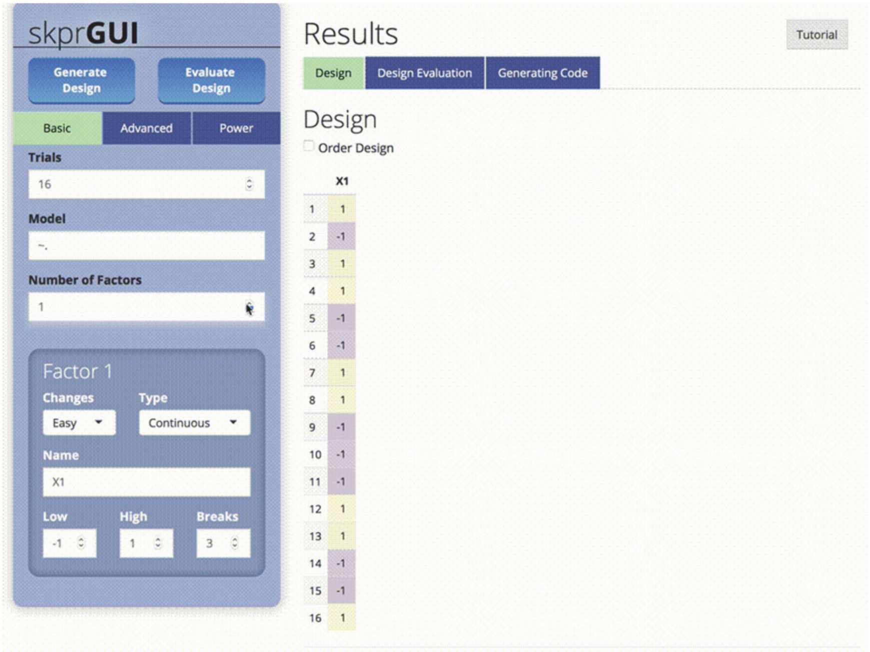
(c) Power values for D-optimal design.

Parameter	Power
(Intercept)	0.945
X1	0.969
X2	0.945
X3	0.945
X4	0.945
X5	0.945
X6	0.945

(d) Power values for Alias-optimal design.

Intro: Why use skpr for Optimal Design of Experiments?

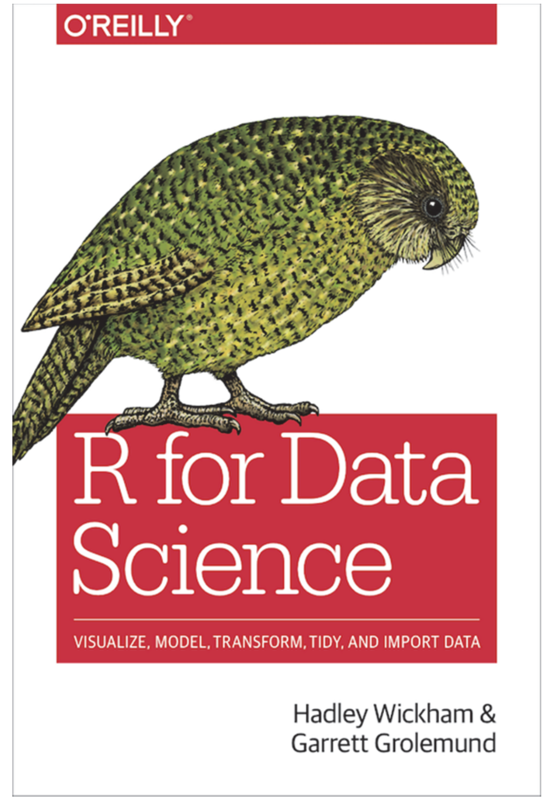
- Designed by testers, for testers: We designed skpr to solve exactly the types of DOE analyses our test community faces.
- End-to-end solution for DOE: Easy interfaces for both design generation and power evaluation are included.
- Open source: It’s free! And you can easily download and audit the underlying code if you have questions about how it works.
- In addition to the scripting interface, it includes an easy web-based graphical user interface to ease the learning curve for working with code.



What you need to know about R (live demo)

R Basics Summary

- Download software in R with `install.packages()`
- Create vectors with the `c()` combine function and using the assignment operator `=`
- Use `for` loops with lists
- Create data frames
- Use R’s formula interface
- Use the R pipe `|>`
- Set a random seed with `set.seed()`
- Generate candidate sets with `expand.grid()`



Learn more using this free e-book:
https://r4ds.had.co.nz/

Using skpr for Design of Experiments

skpr: Core functionality

- **gen_design():** Optimal design generation
- **eval_design():** Parametric power evaluation
- **eval_design_mc():** Monte Carlo power evaluation
- **eval_design_survival_mc():** Monte Carlo power evaluation for survival models (censored data)
- **eval_design_custom_mc():** Framework for the user to implement their own Monte Carlo power evaluation calculations

skpr: Core functionality—what we'll cover in this workshop

- **gen_design():** Optimal design generation
- **eval_design():** Parametric power evaluation
- **eval_design_mc():** Monte Carlo power evaluation
- **eval_design_survival_mc():** Monte Carlo power evaluation for survival models (censored data)
- **eval_design_custom_mc():** Framework for the user to implement their own Monte Carlo power evaluation calculations

skpr: Plotting and functions to assist common workflows

- **skprGUI():** A GUI wrapper to help support limited analyses and introduce users to the software
- **plot_fds():** Generates fraction of design space plots
- **plot_correlations:** Generates correlation plots
- **calculate_power_curves():** Automates the generation and creation of designs across a range of input factors to generate power vs sample/effect size curves

Workflow in skpr

Step 1	Step 2	Step 3
Construct Candidate Set	Generate Optimal Design	Evaluate Statistical Power
expand.grid()	gen_design()	eval_design()
		eval_design_mc()

Step 1 → Step 2 → Step 3

Alternative workflow in skpr

Step 1	→	Step 2
Construct Candidate Set		Automate the Generation and Evaluation of Designs
expand.grid()		calculate_power_curves()

Outline

1. Introduction to Design of Experiments/R/skpr
2. Optimal Design Generation
3. Evaluating Statistical Power
4. skprGUI

Example: Testing Synthetic Aperture Radar image quality

Factors:

- **Aircraft altitude**: 10,000–30,000 ft
- **Aircraft speed**: 450–550 mph
- **Scan mode**: Scan, Spotlight, Strip
- **Environment (hard to change)**: Urban, Desert

Continuous Response: NIIRS image quality (1–9 rating)

Binary Response: Operator can correctly classify aircraft (Yes/No)

Example SAR modes

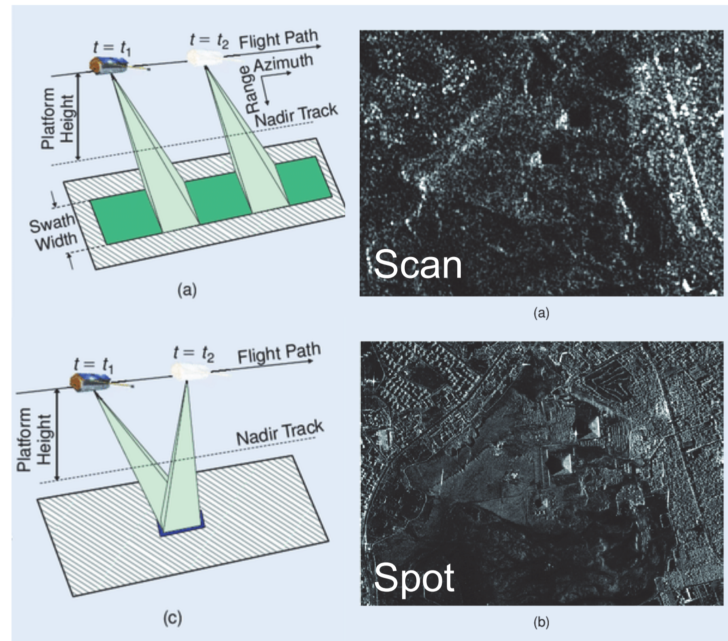


Image: Alberto Moreira et al. (2013). "A Tutorial on Synthetic Aperture Radar." *IEEE Geoscience and Remote Sensing Magazine* 1, No. 1.

NIIRS Rating	Description
0	Interpretability of the imagery is precluded by obscuration, degradation, or low resolution.
1	Detect lines of transportation, either road or rail, but do not distinguish between.
2	Distinguish between forested areas and agricultural fields.
3	Detect multiple wings of large buildings.
4	Detect smokestacks in industrial facilities.
5	Distinguish between a large vertical mast antenna and a large power-transmission tower.
6	Detect cargo on a railroad flatcar.
7	Detect the break between cab and trailer on a tractor-trailer truck.
8	Detect individual rail ties.
9	Distinguish between models of fighter aircraft (e.g., FLANKER B-C, F-15 A-E).

Image/table: Wade Schwartzkopf et al. (2022). "Radar Generalized Image Quality Equation Applied to Capella Open Dataset." *2022 IEEE Radar Conference (RadarConf22).*

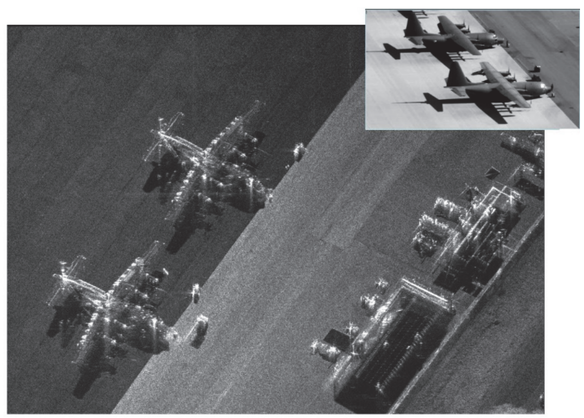


Image: Sandia National Lab
www.sandia.gov/radar/images/ka_band_portfolio.pdf

NIIRS: National Imagery Interpretability Rating Scale

What are examples of constraints?

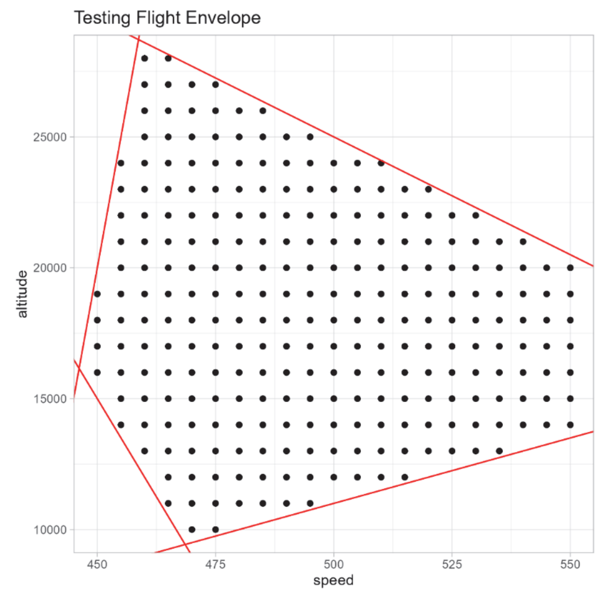
Available Resources

Image source: U.S. State Department

Safety Concerns

Image source: The Dawn Project

Disallowed Combinations



gen_design() – Basic Functionality

Candidate Set

altitude	mode	speed
10000	Scan	450
10000	Scan	500
10000	Scan	550
10000	Spot	450
10000	Spot	500
10000	Spot	550
30000	Scan	450
30000	Scan	500
30000	Scan	550
30000	Spot	450
30000	Spot	500
30000	Spot	550

`sar_test_points`

Model

`~ altitude + mode + speed`

Trials

`10`

Inputs

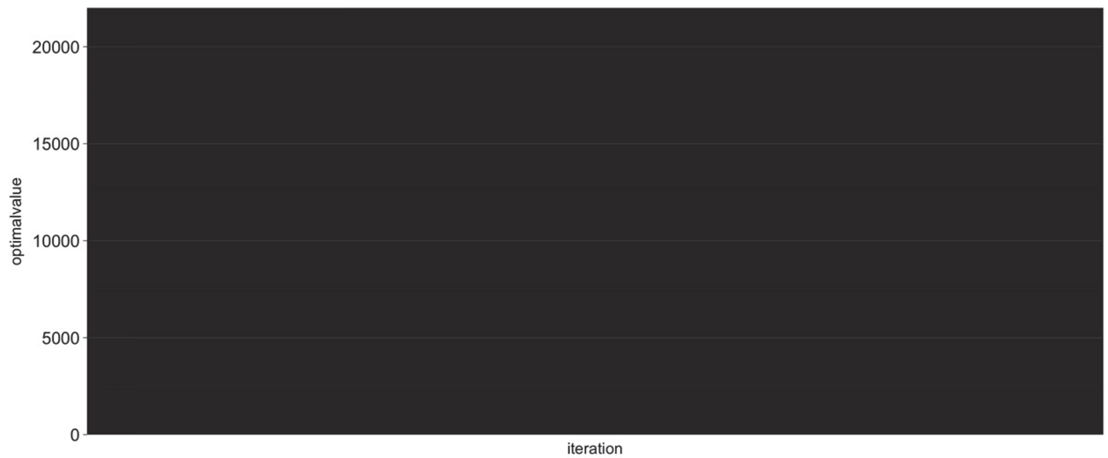
```
gen_design(candidateset = sar_test_points,
           model = ~altitude + mode + speed,
           trials = 10)
```

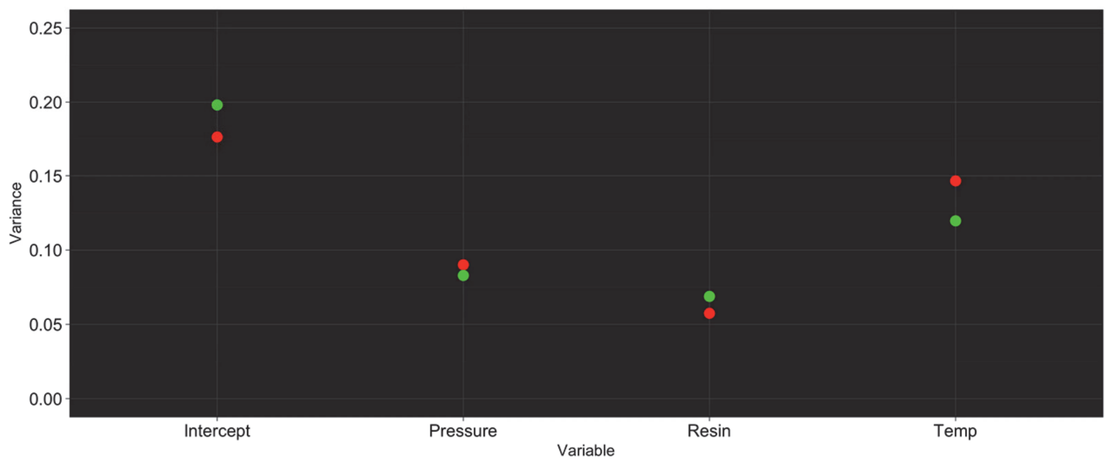
Output

Default is D-Optimal

Run	altitude	mode	speed
1	10000	Spot	450
2	30000	Spot	550
3	30000	Scan	550
4	10000	Scan	550
5	30000	Scan	550
6	30000	Spot	450
7	10000	Scan	450
8	30000	Spot	450
9	10000	Scan	450
10	10000	Spot	550

Designs generated using a point exchange algorithm





Design

Temp	Pressure	Resin
80	2	Slow
80	4	Slow
80	6	Fast
80	2	Fast
100	4	Slow
80	4	Fast
100	4	Fast
100	2	Slow
80	6	Slow
90	6	Slow
80	2	Slow
90	4	Slow

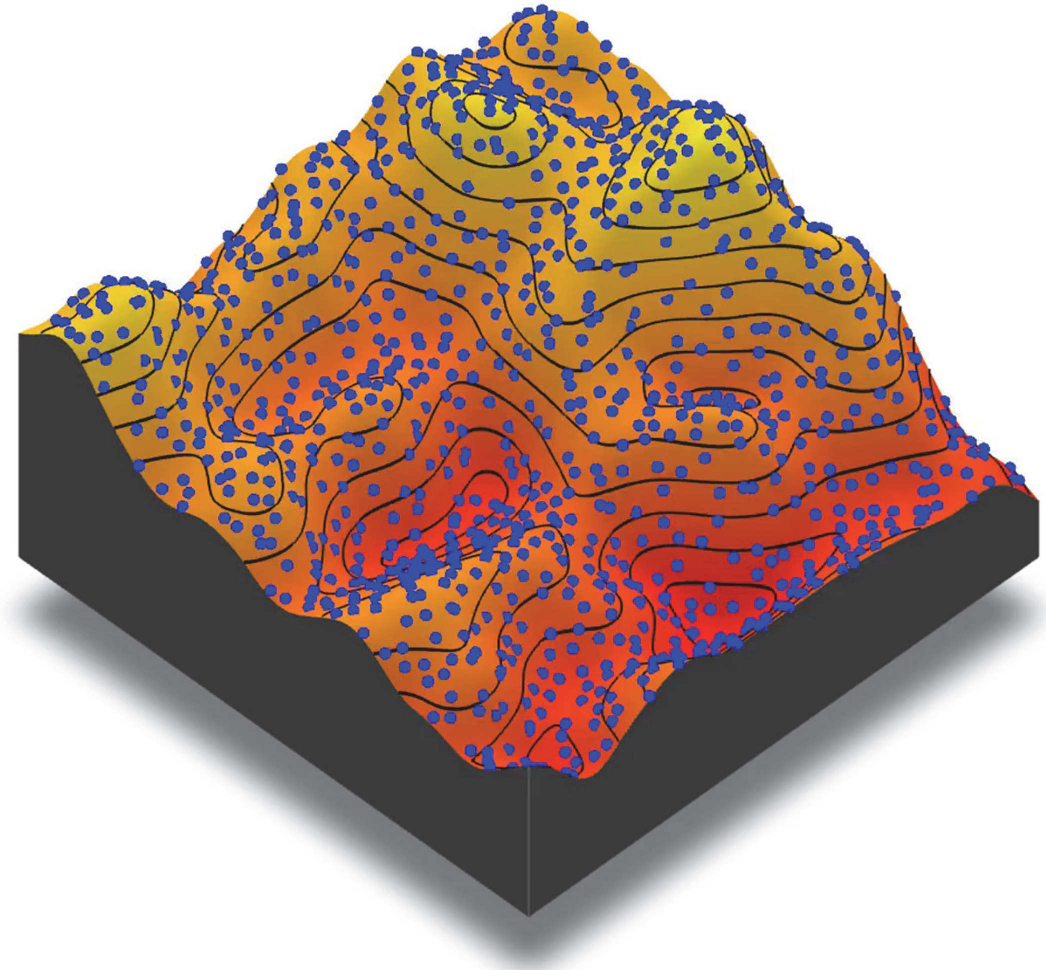
Coded Design

Temp	Pressure	Resin
-1	-1	1
-1	0	1
-1	1	-1
-1	-1	-1
1	0	1
-1	0	-1
1	0	-1
1	-1	1
-1	1	1
0	1	1
-1	-1	1
0	0	1

Candidate Set

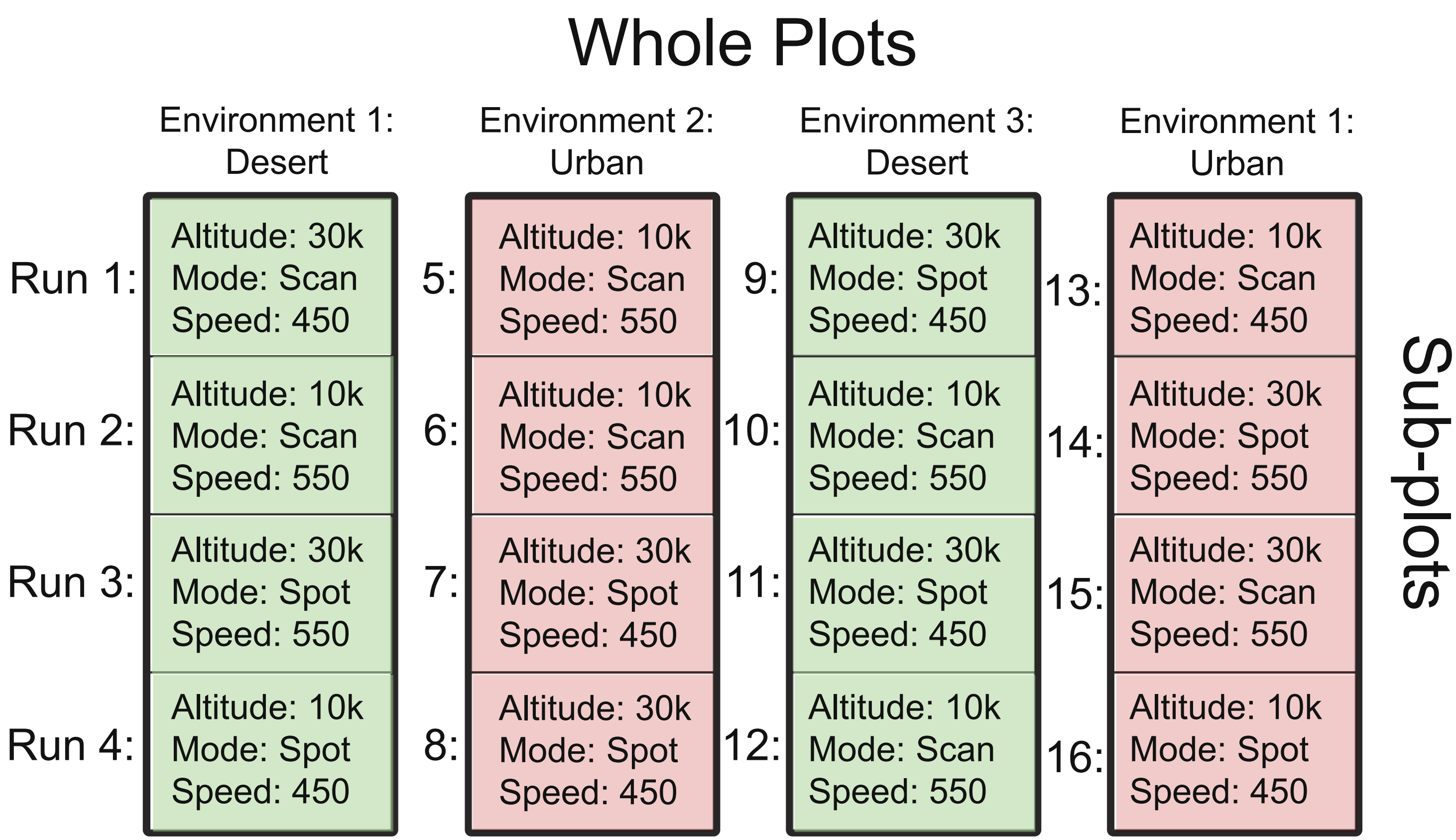
Temp	Pressure	Resin
80	2	Slow
90	2	Slow
100	2	Slow
80	4	Slow
90	4	Slow
100	4	Slow
80	6	Slow
90	6	Slow
100	6	Slow
80	2	Fast
90	2	Fast
100	2	Fast
80	4	Fast
90	4	Fast
100	4	Fast
80	6	Fast
90	6	Fast
100	6	Fast

Optimizing many random designs improves chance of finding true optimum



Generating Designs (live demo)

Split-plot designs for hard-to-change factors



Creating split-plot designs in skpr

- In skpr, split-plot designs are created in layers: first by generating an optimal design for the hardest to change factors, and then generating designs for the easier-to-change factors, with the harder-to-change design fixed.
- Any depth of split-plot structure is possible in skpr using this method (split-split-plot, split-split-split-plot, etc.).
- The user can specify their own block sizes or have skpr automatically generate blocks to be as balanced as possible.

	target_env
1	urban
2	desert
3	urban
4	urban
5	desert
6	urban
7	desert
8	desert

Passed in via argument `splitplotdesign`

Block Size

4
4
4
4
4
4
4
4

	target_env	altitude	speed	mode
1.1	urban	12000	515	scan
1.2	urban	28000	460	scan
1.3	urban	20000	550	spotlight
1.4	urban	18000	450	spotlight
2.1	desert	18000	490	scan
2.2	desert	20000	550	strip
2.3	desert	10000	470	strip
2.4	desert	10000	470	spotlight
3.1	urban	10000	470	scan
3.2	urban	12000	515	spotlight
3.3	urban	20000	550	scan
3.4	urban	19000	450	strip
4.1	urban	28000	460	spotlight
4.2	urban	20000	550	scan
4.3	urban	10000	470	spotlight
4.4	urban	18000	450	scan
5.1	desert	14000	550	scan
5.2	desert	28000	460	scan
5.3	desert	23000	520	strip
5.4	desert	10000	470	strip
6.1	urban	20000	550	spotlight
6.2	urban	24000	510	scan
6.3	urban	14000	550	strip
6.4	urban	28000	460	strip
7.1	desert	14000	550	spotlight
7.2	desert	10000	470	scan
7.3	desert	28000	460	spotlight
7.4	desert	18000	495	strip
8.1	desert	28000	460	strip
8.2	desert	14000	550	strip
8.3	desert	24000	510	spotlight
8.4	desert	16000	450	spotlight

Outline

1. Introduction to Design of Experiments/R/skpr
2. Optimal Design Generation
3. Evaluating Statistical Power
4. skprGUI

What is statistical power?

Statistical power: the probability that your experiment will find an effect (reject the null hypothesis) if one does exist (the null hypothesis is false).

The acceptable Type-I error probability α is set by the tester ahead of time, while statistical power (1- β) is calculated from the design and other inputs.

Some inputs that affect power:

1. **Design size** (number of runs)
2. **Design quality** (optimality)
3. **Type-I error rate** (lower acceptable Type-I error probability = lower power)
4. **Effect size** (typically given as a signal-to-noise ratio)
5. **Analysis methods** (e.g., an exact binomial test versus logistic regression)

Type-I Error α
(probability of false positive)

"The new radar indicates incoming balloons!"

Type-II Error β
(probability of false negative)

"The new radar doesn't detect any balloons entering our airspace."

Why 80% power?

- The selection of 80% power as a goal is not arbitrary: It is a decision about how we weigh the different types of statistical risks involved.
- **Example**: By setting the acceptable Type-I error rate to 0.05 (95% confidence) and the acceptable Type-II error rate to 0.20 (80% power), we are saying we believe a false positive (e.g., *the results say the system meets requirements when in reality it does not*) is 4x worse than a false negative (e.g., *the results say the system fails when in reality it meets its requirements*).
- **Example**: By setting the acceptable Type-I error rate to 0.20 (80% confidence) and the acceptable Type-II error rate to 0.20 (80% power), we are saying we consider a false negative and a false positive to be equally important.

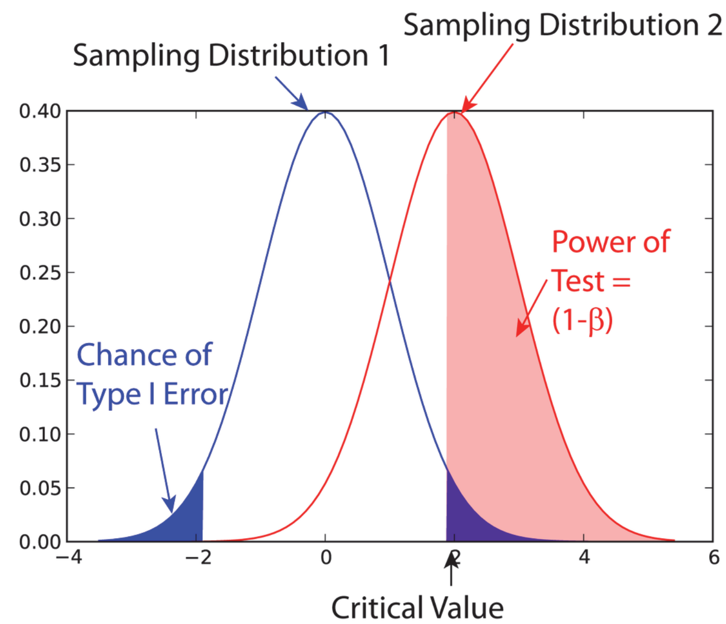
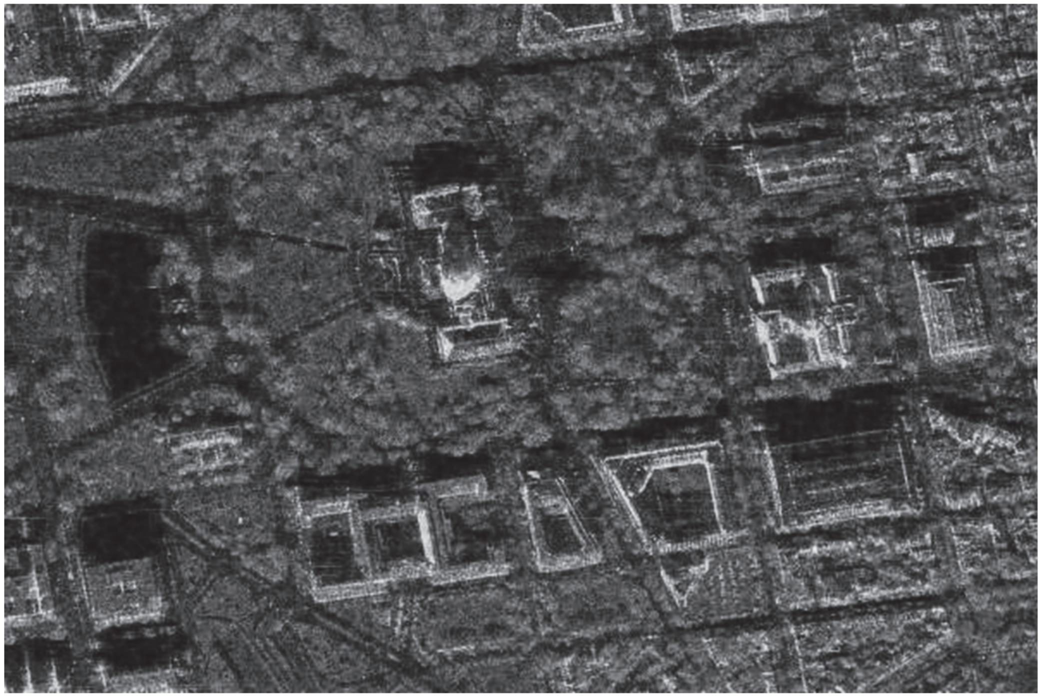


Image: https://commons.wikimedia.org/wiki/File:Statistical_test,_significance_level,_power.png

What is effect size?

- For linear models, an effect size is a signal-to-noise ratio (SNR): the ratio of a meaningful effect you are trying to detect with the expected intrinsic run-to-run variability of the response.
- Example: NIIRS rating
- For non-Gaussian generalized linear models, your effect size is given as the following:
  - Binomial: Odds ratios (or equivalently, a low and high probability which are converted to odds ratios)
  - Poisson: Number of events (e.g., one false classification per scan vs four false classifications)
  - Exponential: Rates (e.g., one failure per hundred operator hours vs four failures per hundred operating hours)
- Example: SAR imagery correct classification probability

NIIRS 5*



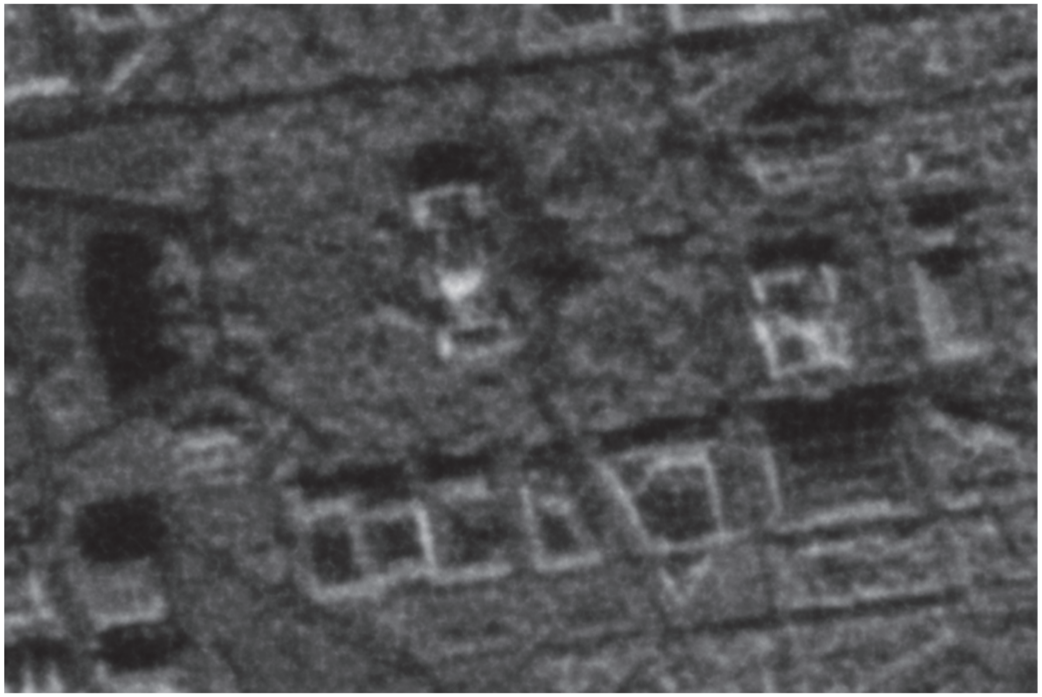
SAR image from Sandia National Laboratories
* Value assigned as an example: no actual NIIRS analysis was performed

NIIRS: National Imagery Interpretability Rating Scale

What is effect size?

- For linear models, an effect size is a signal-to-noise ratio (SNR): the ratio of a meaningful effect you are trying to detect with the expected intrinsic run-to-run variability of the response.
- Example: NIIRS rating
- For non-Gaussian generalized linear models, your effect size is given as the following:
  - Binomial: Odds ratios (or equivalently, a low and high probability which are converted to odds ratios)
  - Poisson: Number of events (e.g., one false classification per scan vs four false classifications)
  - Exponential: Rates (e.g., one failure per hundred operator hours vs four failures per hundred operating hours)
- Example: SAR imagery correct classification probability

NIIRS 4*



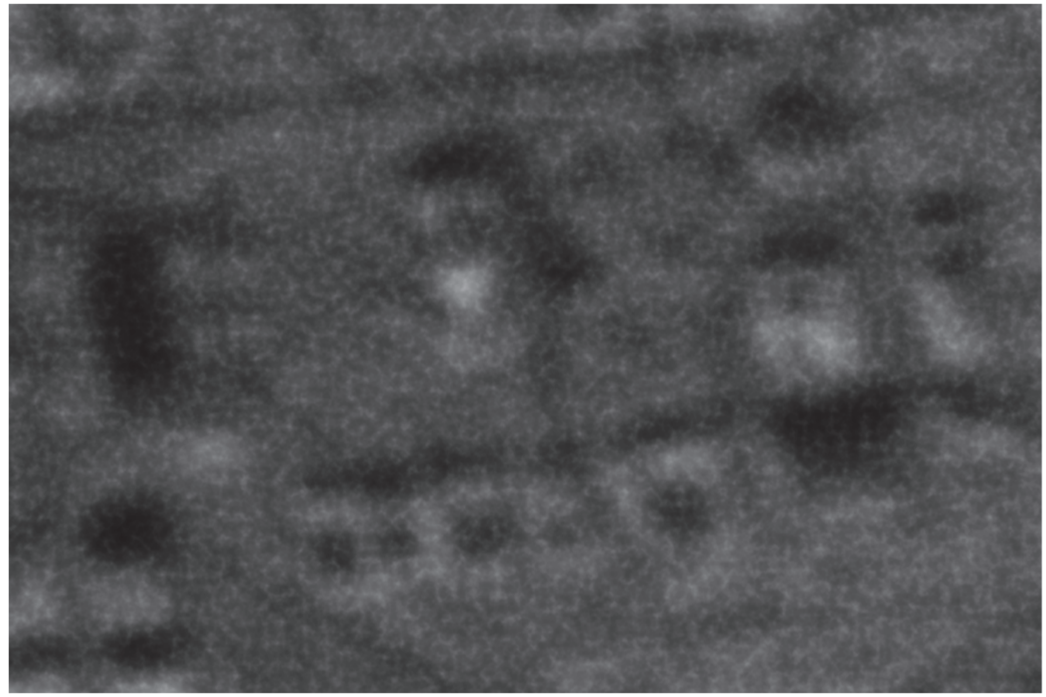
SAR image from Sandia National Laboratories (modified for example)
* Value assigned as an example: no actual NIIRS analysis was performed

NIIRS: National Imagery Interpretability Rating Scale

What is effect size?

- For linear models, an effect size is a signal-to-noise ratio (SNR): the ratio of a meaningful effect you are trying to detect with the expected intrinsic run-to-run variability of the response.
- Example: NIIRS rating
- For non-Gaussian generalized linear models, your effect size is given as the following:
  - Binomial: Odds ratios (or equivalently, a low and high probability which are converted to odds ratios)
  - Poisson: Number of events (e.g., one false classification per scan vs four false classifications)
  - Exponential: Rates (e.g., one failure per hundred operator hours vs four failures per hundred operating hours)
- Example: SAR imagery correct classification probability

NIIRS 3*



SAR image from Sandia National Laboratories (modified for example)
* Value assigned as an example: no actual NIIRS analysis was performed

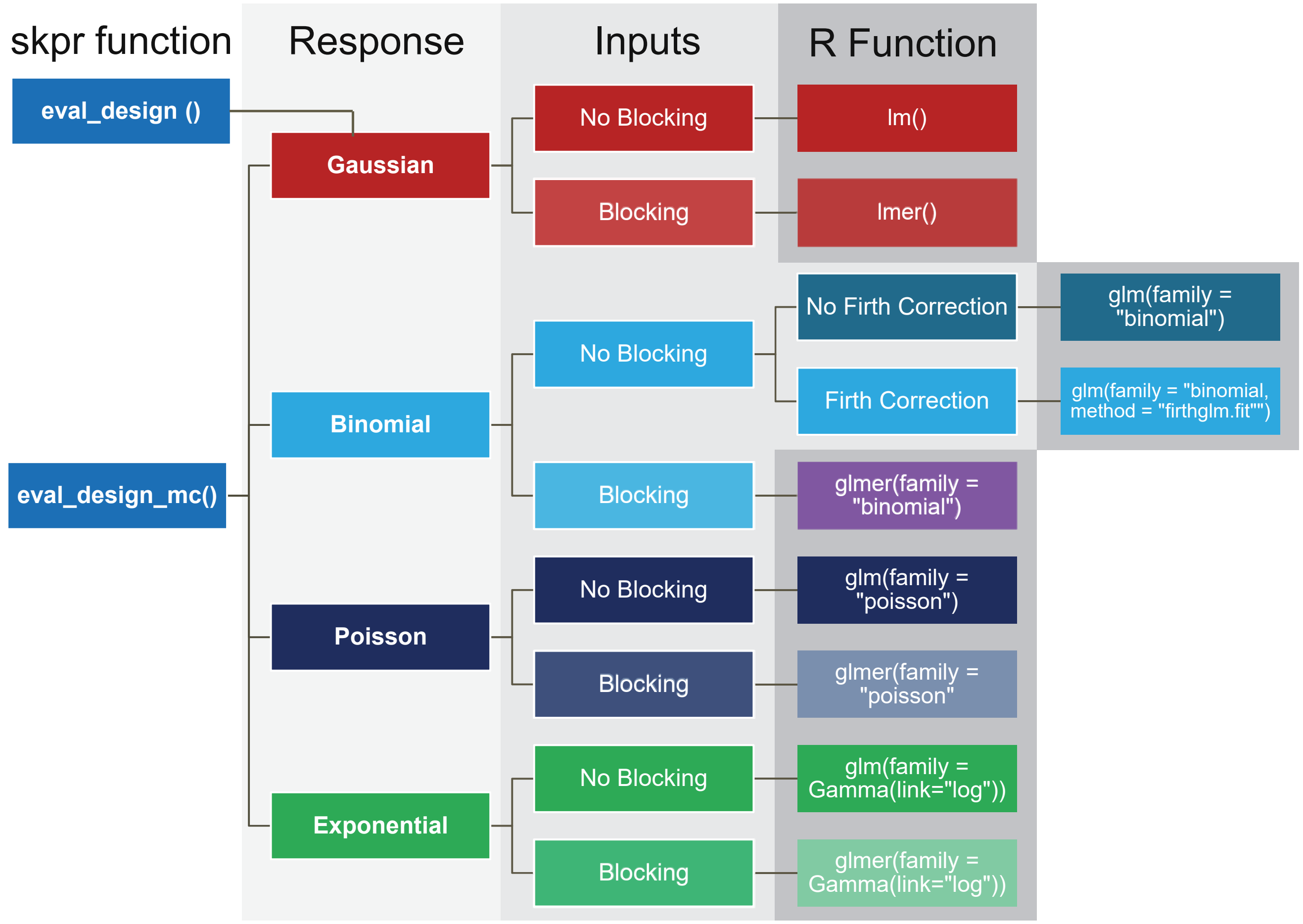
NIIRS: National Imagery Interpretability Rating Scale

Parameter power and effect power refer to different analysis methods

Parameter Power	Effect Power
• Power when fitting a regression model and estimating significance for each model term • For categorical factors with dummy encoding, regression allows you to determine which factor level is significant, compared to a reference level (e.g., Mode 2 has a significant performance impact on the system compared to Mode 1, while Mode 3 does not)	• Power when performing an ANOVA or a Likelihood Ratio Test (comparing two models, one with and one without each term) and estimating if the difference between the two models is significant • For categorical factors, this type of analysis does not provide you with information on what specific levels are or are not significant—only that the term itself has a significant effect on the response

2-for-1 deal: A power analysis effectively is your analysis plan

- While certain design metrics (like D-optimality) are independent of the planned analysis methods, **statistical power is not!**
- Your power values depend on your analysis method, so you need to make sure the assumptions you make during the power analysis match what you plan to do with the actual data.
- Thankfully, you don't have to try to memorize this chart: skpr prints the R function/analysis method used along with the power output.



Some statistical methods are more powerful than others

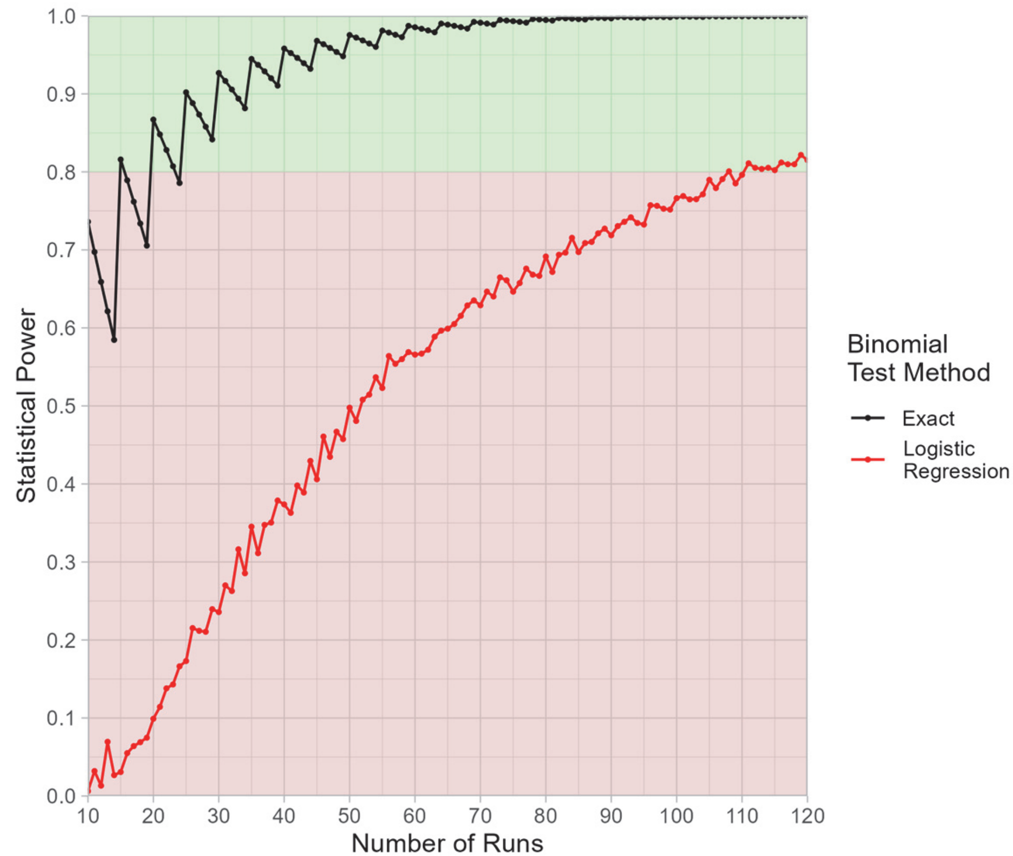
- Using one method to calculate power and a different method for your actual analysis can lead to inaccurate power results.
- Inaccurate power results may lead to oversized or undersized tests, wasting resources, time, and money.

Example: Logistic regression versus exact binomial test

- **Base Probability**: 0.75
- **Threshold Probability**: 0.9
- **Alpha** = 0.2

Sample size required for 80% power:

- **Exact binomial test**: 15 runs
- **Logistic regression**: 110 runs



eval_design() – Basic Functionality

Design

altitude	speed	mode
10000	450	scan
30000	550	scan
10000	550	scan
10000	550	spotlight
10000	450	spotlight
30000	450	strip
30000	550	strip
10000	550	strip
30000	450	spotlight
10000	450	strip

`sar_design`

Model

`Y ~ altitude + speed + mode`

Alpha

`0.2`

Inputs

```
eval_design(design = sar_design,
            model = ~altitude + speed + mode
            alpha = 0.2)
```

Output

```
    parameter            type     power
1 (Intercept)    effect.power 0.9329381
2    altitude    effect.power 0.9329381
3       speed    effect.power 0.9329381
4        mode    effect.power 0.6886106
5 (Intercept) parameter.power 0.9329381
6    altitude parameter.power 0.9329381
7       speed parameter.power 0.9329381
8       mode1 parameter.power 0.7389917
9       mode2 parameter.power 0.7389917
============Evaluation Info============
• Alpha = 0.2 • Trials = 10 • Blocked = FALSE
• Evaluating Model = ~altitude + speed + mode
• Anticipated Coefficients = c(1, 1, 1, 1, -1)
```

Evaluating Designs (live demo)

eval_design_mc(): No need to approximate when you can simulate

- Calculating power for anything more complex than a non-blocked design with a normal response requires making approximations.
- Those approximations tend to fail at low numbers of runs (where power matters most).
- In many cases, no approximations exist.

eval_design_mc() – Basic Functionality

Design

altitude	speed	mode
10000	450	scan
30000	550	scan
10000	550	scan
10000	550	spotlight
10000	450	spotlight
30000	450	strip
30000	550	strip
10000	550	strip
30000	450	spotlight
10000	450	strip

`sar_design`

Model: `Y ~ altitude + speed + mode`

Alpha: `0.2`

nsim: `10000`

glmfamily: `gaussian`

Inputs

```
eval_design(design = sar_design,
            model = ~altitude + speed + mode
            alpha = 0.2
            nsim = 10000,
            glmfamily = "gaussian")
```

Output

```
    parameter            type     power
1 (Intercept)    effect.power 0.9329381
2    altitude    effect.power 0.9329381
3       speed    effect.power 0.9329381
4        mode    effect.power 0.6886106
5 (Intercept) parameter.power 0.9329381
6    altitude parameter.power 0.9329381
7       speed parameter.power 0.9329381
8       mode1 parameter.power 0.7389917
9       mode2 parameter.power 0.7389917
============Evaluation Info============
• Alpha = 0.2 • Trials = 10 • Blocked = FALSE
• Evaluating Model = ~altitude + speed + mode
• Anticipated Coefficients = c(1, 1, 1, 1, -1)
```

Outline

1. Introduction to Design of Experiments/R/skpr
2. Optimal Design Generation
3. Evaluating Statistical Power
4. skprGUI

skprGUI (live demo)

Summary: A DOE Checklist

1. Determine if you’re more interested in characterization (D-optimal), prediction (I-optimal), or screening (Alias-optimal).
2. Increase the number of repeat design searches when your factor space is complex (large or has disallowed combinations).
3. Generate power curves to understand the trade-off space between model complexity, effect size, and sample size.
4. Use Monte Carlo methods for your final estimate of power for more robust and accurate estimates, especially if you are working with unbalanced designs, generalized linear models, or models with blocking/random effects.
5. Once you have your power estimates, run Monte Carlo power calculations with the null hypothesis to ensure Type-I error is not inflated (estimated power is approximately equal to alpha).
6. **Ensure that you know how your power estimates were calculated, and use those same methods in your final analysis!**

Resources

- Install **R:**
  https://www.r-project.org/
- Install **RStudio:**
  https://www.posit.co/downloads/
- Read **R for Data Science:**
  https://r4ds.had.co.nz/
- **skpr** GitHub (source code + latest releases):
  https://www.github.com/tylermorganwall/skpr
- **skpr** Journal of Statistical Software paper:
  https://www.jstatsoft.org/article/view/v099i01
- **skpr** Shiny app:
  https://www.testscience.org

Extra Slides

eval_design_survival_mc: Censored power calculations

Design
`sar_design`

Model
`Y ~ size + type + temp`

Alpha
`0.2`

distribution
`exponential`

Inputs

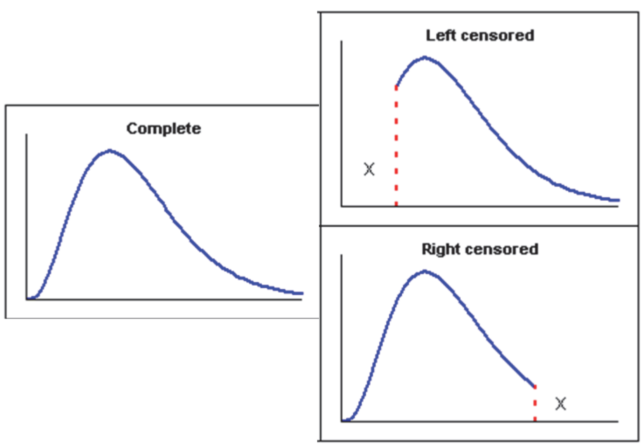
Censor Point
`5`

Censor Type
`right`

```
eval_design_survival_mc(design = sar_design,
          model = ~size + type + temp,
          alpha = 0.2,
          censorpoint=5,
          censortype="right",
          distribution= "exponential")
```

Output

	parameters	type	power
1	(Intercept)	parameter.power.mc	0.930
2	size1	parameter.power.mc	0.702
3	size2	parameter.power.mc	0.267
4	type1	parameter.power.mc	0.780
5	temp	parameter.power.mc	0.799



Optimality Criteria

1. D-optimal designs minimize $|(X^\top X)^{-1}|$, or equivalently maximize $|X^\top X|$, the determinant of the information matrix (Atkinson and Donev 1992).

2. I-optimal designs minimize the average prediction variance over the design space, $\mathrm{tr}[(X^\top X)^{-1}M]$.

3. Alias-optimal designs minimize the trace of the sum of squares of the alias matrix A, $\mathrm{tr}(A^\top A)$ (where $A = (X^\top X)^{-1}X^\top X_2$ and X_2 is the model matrix representing the interaction columns), while simultaneously ensuring the D-optimality does not drop below a certain user-defined threshold. This function can generate designs with a favorable aliasing structure, which enables the user to conduct screening experiments for active effects (Jones and Nachtsheim 2011b).

4. A-optimal designs minimize $\mathrm{tr}[(X^\top X)^{-1}]$, the trace of the inverse of the information matrix. This criterion results in minimizing the average variance of the estimates of the regression coefficients (Atkinson and Donev 1992).

5. G-optimal designs minimize the maximum entry in the diagonal of $X(X^\top X)^{-1}X^\top$, the hat matrix. This minimizes the maximum variance of the predicted values (Atkinson and Donev 1992).

6. E-optimal designs maximize the minimal eigenvalue of the information matrix $X^\top X$, which minimizes the worst-case variance of any linear combination of estimated coefficients (Atkinson and Donev 1992).

7. T-optimal designs maximize the trace of the information matrix, $\mathrm{tr}(X^\top X)$ (Atkinson and Donev 1992).

8. Custom-optimal designs maximize a function of the design's model matrix defined by the user, $f(X)$.

Blocked/Split-Plot Designs

For these designs, the parameter estimate is no longer the ordinary least squares (OLS) estimator,

$$\hat{\beta} = (X^\top X)^{-1}X^\top Y,$$

but rather the generalized least squares (GLS) estimate (also named the feasible GLS estimator) (Aitken 1936):

$$\hat{\beta} = (X^\top V^{-1}X)^{-1}X^\top V^{-1}Y,$$

where V is the covariance matrix of the response vector, which **skpr** calculates automatically from the blocking structure of a design. The D-optimal condition is no longer to maximize the determinant of the information matrix $X^\top X$, but rather the determinant of $X^\top V^{-1}X$. This extends to other optimal conditions; the insertion of the V^{-1} in the optimality criteria is as follows in **skpr**:

1. D-optimal: maximize $|X^\top V^{-1}X|$.
2. I-optimal: minimize $\mathrm{tr}[(X^\top V^{-1}X)^{-1}M]$.
3. A-optimal: minimize $\mathrm{tr}[(X^\top V^{-1}X)^{-1}]$.
4. G-optimal: minimize the maximum diagonal element of $X(X^\top V^{-1}X)^{-1}X^\top V^{-1}$.
5. E-optimal: maximize the minimal eigenvalue of $X^\top V^{-1}X$.
6. T-optimal: maximize $\mathrm{tr}(X^\top V^{-1}X)$.
7. Custom-optimal: maximize a user-defined function $f(X, V^{-1})$.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
14-03-2023

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
N/A

4. TITLE AND SUBTITLE
DATAWorks 2023: "Introduction to Design of Experiments in R: Generating and Evaluating Designs with skpr"

5a. CONTRACT NUMBER
Separate Contract

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER
C9117

5e. TASK NUMBER
C9117

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Tyler T Morgan-Wall (OED)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
730 East Glebe Road
Alexandria, Virginia 22305

8. PERFORMING ORGANIZATION REPORT NUMBER
D-33397

H 2023-000047

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
730 East Glebe Road
Alexandria, Virginia 22305

10. SPONSOR/MONITOR'S ACRONYM(S)
IDA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
This OED Draft has not been approved by the sponsor for distribution and release. Reproduction or use of this material is not authorized without prior permission from the responsible IDA Division Director.

13. SUPPLEMENTARY NOTES
Dr. Tyler T Morgan-Wall, Project Leader

14. ABSTRACT
This workshop instructs attendees on how to run an end-to-end optimal Design of Experiments workflow in R using the open source skpr package. This workshop is split into two sections: optimal design generation and design evaluation. The first half of the workshop provides basic instructions how to use R, as well as how to use skpr to create an optimal design for an experiment: how to specify a model, create a candidate set of potential runs, remove disallowed combinations, and specify the design generation conditions to best suit an experimenter's goals. The second half of the workshop covers design evaluation with skpr: how to determine if an experimental design is adequate for the test at hand. The workshop provides information on how to perform power calculations and evaluate other design properties that affect design quality. This also includes instruction on how to generate fraction of design space plots and correlation plots.

15. SUBJECT TERMS
Design of Experiments, DOE, R programming, R language, Optimal Design, skpr

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	SAR	58	Dr. Tyler T. Morgan-Wall
					19b. TELEPHONE NUMBER *(Include area code)* 7038452095